



⑬ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Patentschrift**
⑩ **DE 198 11 593 C 1**

⑳ Aktenzeichen: 198 11 593.8-31
㉔ Anmeldetag: 17. 3. 98
㉕ Offenlegungstag: -
㉖ Veröffentlichungstag
der Patenterteilung: 6. 5. 99

㉑ Int. Cl.⁶:
H 04 N 1/44

G 06 K 5/00
G 06 K 5/00
G 07 C 11/00
G 09 C 1/14
H 04 K 1/00
H 04 L 9/32

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

⑥⑤ Innere Priorität:

197 56 792. 4 19. 12. 97
198 11 111. 8 13. 03. 98

⑦③ Patentinhaber:

V + S Datentechnik und Software GmbH, 81545
München, DE

⑦④ Vertreter:

Leonhard und Kollegen, 80331 München

⑦② Erfinder:

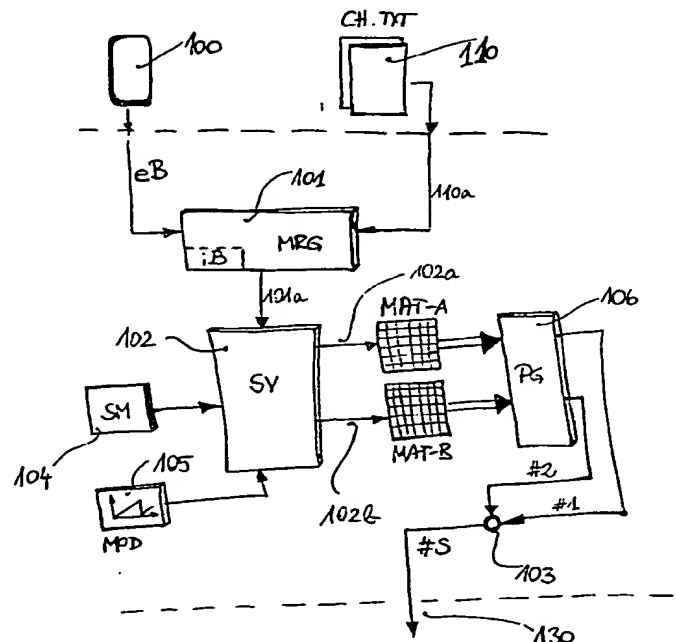
Schnoor, Ernst Erich, 81545 München, DE

⑥⑥ Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:

ZHENG. Y. u.a.: HAVAL-a one-way hashing
algorithm
with variable length of output, In: Advances in
Cryptography-AUSCRYPTC92, Berlin, Springer-Verlag
1993, S. 83-104;
ZEMOR-G.: Hash functions and graphs with large
girth, In: Advances in Cryptology-Eurocrypt'91,
Berlin, Springer-Verlag 1991, S. 508-511;

⑤④ Matrixgesteuerte Hashfunktion

⑤⑦ Vorgeschlagen wird ein Verfahren zum Berechnen eines Kontrollwortes (Hashwert) einer Text- oder Informationsdatei, das mit der Datei zusammen übertragen wird, um dem Empfänger die Integrität der mitübertragenen Datei oder die richtige Herkunft (Authentizität) der Datei zu signalisieren. Eine erste Symbolverteilung (SV, 102) wird mit einem ersten, in seiner Länge gegenüber der Gesamtlänge der Datei (110) kurzen Block von Zeichen (B1) der Datei in einer vorgegebenen Dimension aufweisenden Matrix (MAT-A, MAT-B) vorgenommen, so daß jedes Element der Matrix ein anderes Symbol aus einem Symbolvorrat (SM) enthält. Die erste Symbolverteilung in der Matrix wird positionsgewichtet (106, PG) zu einem ersten Teil-Hashwert (h_1) zusammengefaßt. Eine geänderte Symbolverteilung (SV, 102) wird mit einem weiteren, der Länge des ersten Blocks entsprechenden zweiten Block (B2) von Zeichen derselben Datei (110) in derselben Matrix (MAT-A) oder einer zweiten Matrix (MAT-B) vorgenommen. Die geänderte Symbolverteilung wird erneut positionsgewichtet (106), um einen weiteren Teil-Hashwert (h_2) zu errechnen. Beide Teil-Hashwerte (h_1, h_2) werden gleich oder später nach Berechnen mehrerer oder aller Teil-Hashwerte zu einem abschließenden Hashwert (#S) zusammengefaßt. Die vorgeschlagene Hashfunktion ermöglicht es, eine Datei beliebiger Länge möglichst eindeutig einem Hashwert zuzuordnen, so daß es praktisch unmöglich wird, Dateien gezielt so zu ändern, daß sie denselben Hashwert ergeben.



DE 198 11 593 C 1

Die Erfindung befaßt sich mit einer Einweg-Hashfunktion, die in der Kryptografie selbst keine Protokolle bildet und auch keine chiffrierten Daten erzeugt, die übertragen werden sollen, sondern als grundlegender Baustein eine Art Fingerabdruck einer zumeist als Klartext vorliegenden lesbaren Textdatei (Dokument, Information, Vertrag, Nachricht, Auftrag) erzeugt. Mit dem durch die Hashfunktion gebildeten Hashwert kann ein Vergleich stattfinden, ob von derselben Datei ausgegangen worden ist oder ob eine nichtverschlüsselte Datei auf dem Daten-Übertragungsweg geändert worden ist oder mit Fehlern versehen wurde. Dann, wenn nicht dieselben Hashwerte aufgrund der Anwendung derselben Hashfunktion auf zwei verschiedene Dateien herauskommen, sind diese beiden verschiedenen Dateien nicht gleich. Das kann in der Kryptoanalyse auf zwei Weisen verwendet werden. Es können zwei Originale an verschiedenen Stellen verglichen werden, wenn nur der Hashwert der einen Datei an den anderen Ort zum Empfänger übermittelt wird und dort von der zu vergleichenden, nicht übertragenen Datei ein Hashwert berechnet wird. Das kann aber auch so verwendet werden, daß eine mit einem Hashwert parallel übermittelte Datei am Empfängerort daraufhin überprüft werden kann, ob beim Anwenden der dem Empfänger bekannten Hashfunktion auf die empfangene Datei derselbe Hashwert herauskommt, der mit der übertragenen Datei übermittelt wurde. Letztere Anwendung der Hashfunktion dient der Nachrichten-Authentifizierung oder Integrität, während Erstere einen Vergleich ermöglicht, ähnlich dem eingangs erwähnten Vergleich von Fingerabdrücken.

Es liegt in der Natur der Sache, daß eine "Einweg-Funktion" tatsächlich nur einen Hinweg hat und der Rückweg praktisch verschlossen ist. Wird eine Klartext-Datei der eingangs genannten Art mit einer Hashfunktion bearbeitet, so kann aus dem sich ergebenden Hashwert die Datei nicht rekonstruiert werden. Die Umkehrfunktion, die bei der Übertragung von verschlüsselten Daten so wichtig ist, findet bei der Hashfunktion also keine Anwendung. Es werden vielmehr Klartext-Dateien übertragen, die von jedermann gelesen werden können, aber nicht geändert werden können, ohne diese Änderungen dem Empfänger der Datei, der einen Vergleich von Hashwerten vornimmt, sichtbar zu machen.

Eine Hashfunktion der beschriebenen Art wird von ZEMOR-G. "Hashfunctions and graphs with large girths" in Advances in Cryptology, Eurocrypt 1991, Springer-Verlag 1991, Seiten 508 bis 511, beschrieben; diese Funktion wandelt große Dateien (mehrere Megabyte) in die Signatur darstellende wenige hundert Bit um. Je größer die Dateien aber werden, desto mehrdeutiger werden die vorgegebenen Bits der Signatur.

Das Problem der Erfindung liegt darin, eine Hashfunktion vorzuschlagen, die es ermöglicht eine Datei beliebiger Länge möglichst eindeutig einem Hashwert zuzuordnen, so daß es praktisch unmöglich wird, Dateien gezielt so zu ändern, daß sie denselben Hashwert ergeben.

Dazu schlägt die Erfindung das Verfahren gemäß Anspruch 1 und das Verfahren gemäß Anspruch 10 vor, wobei das Verfahren gemäß Anspruch 10 den Empfänger betrifft, der eine nicht chiffrierte Datei empfängt und einen empfangenen Hashwert mit einem selbstgenerierten Hashwert aus der empfangenen (oder beim Empfänger vorhandenen) Datei vergleicht, um sich Sicherheit über die Integrität oder die Authentizität der Datei zu verschaffen.

Die Erfindung löst sich von dem Gedanken, eine feste Länge des Hashwertes anzugeben, vielmehr wird der Hashwert um so länger oder um so größer, je länger die Datei ist, die von der Hashfunktion bearbeitet wird, die also die Eingangsgröße der erfindungsgemäßen Hashfunktion ist. Es ist im Stand der Technik zwar eine Einweg-Hashfunktion schon beschrieben, die eine variable Länge hat; diese Hashfunktion heißt HAVAL, vgl. Bruce Schneier, "Angewandte Kryptologie", Addison-Wesley, Ziffer 18.9, Seite 508 oder Zheng, Pieprzyk, Scherrey, "HAVAL - a one-way hashing algorithm with variable length of output" - Abstract of: Advances in Cryptology, AUSCRIPT 92, Berlin, Springer-Verlag, 1993, Seiten 83-104. Diese HAVAL-Funktion arbeitet mit variabler Rundenzahl von 3 bis 5 (mit jeweils 16 Schritten) und kann Hashwerte der Länge 128, 160, 192, 224 oder 256 Bit erzeugen. Ein gleitender, stetig mit der Länge der Datei, die der Hashfunktion unterworfen wird, wachsender Hashwert ist diese Funktion nicht, dagegen erlaubt die Erfindung ein größer werdendes Kontrollwort (Hashwert), wenn die Dateien größer werden. Bereits durch diese grundlegende Überlegung kann erfindungsgemäß eine Datei möglichst eindeutig einem Hashwert zugeordnet werden, wobei ersichtlich ist, daß eine größere Datei für eine eindeutige Abbildung in einem Hashwert auch einen größeren Hashwert benötigt. Umgekehrt kann eine Hashfunktion, wie SNEFRU, die 128 oder 256 Bit als Hashwert mit fester Länge erzeugt, bei immer größeren Dateien in die Probleme geraten, daß immer mehr verschiedene (gezielt geänderte) Dateien in denselben Hashwert abgebildet werden. Gezielte Änderungen, die an mehreren Stellen einer Datei vorgenommen werden, können so nicht mehr durch die Hashfunktion erkannt werden und das eröffnet größere Möglichkeiten für Fälschungen in übertragenen Klartext-Dateien, die dem Empfänger als zwar geänderte Datei übermittelt werden, aber bei Erproben der empfangenseitigen bekannten Hashfunktion denselben Hashwert liefert, den diejenige Datei geliefert hat, die noch nicht verfälscht war.

Im Kern der Erfindung liegt es, die Datei, die der Hashfunktion unterworfen wird in kurze Blöcke aufzuteilen, so beispielsweise Blocklängen von > 2 byte, bevorzugt in der Größenordnung von 30 bis 50 byte. Jeder Block dient einer Symbolverteilung von Symbolen in einer vorgegebenen Matrix mit einer definierten Dimension. Es können auch mehrere Matrizen verwendet werden, ebenso wie mehrdimensionale Matrizen Einsatz finden können. Die Symbole werden auf die Elemente der Matrix so verteilt, daß in allen Elementen der Matrix kein Symbol doppelt vorkommt. Diese Verteilung der Symbole, was voraussetzt, daß mehr oder zumindest gleich viel Symbole als Symbolvorrat zur Verfügung stehen, als Elemente der Matrix vorhanden sind, wird gesteuert von dem ersten Ausschnitt (Block) der Klartext-Datei, die im folgenden nur als "Datei" bezeichnet werden soll. Der Ausschnitt ist der erste kurze Block und er steuert die Symbolverteilung. Diese Symbolverteilung wird erfindungsgemäß positionsgewichtet und bildet einen ersten Teil-Hashwert, sei es durch Addition, Subtraktion oder eine sonstige Rechenfunktion. Die Positionsgewichtung kann so erfolgen, daß jedes Symbol mit dem Platz in der Matrix gewichtet wird, an dem es durch den ersten kurzen Block gesteuert gelange. Das kann eine aufsteigende numerische Variable und/oder eine fortlaufende Ziffer von Reihen sein, die durch einen später zu erläuternden mathematischen Reihengenerator zur Verfügung gestellt wurde.

Ein weiterer Teil-Hashwert wird für den nächsten kurzen Block, der der Länge des ersten Blocks entspricht und den folgenden Ausschnitt der Klartext-Datei darstellt, auf gleiche Weise ermittelt, wie der erste Teil-Hashwert. Auch hier

wird eine Symbolverteilung vorgenommen, die aber jetzt anders ist, weil die Längengröße (der zweite Ausschnitt aus der Datei) andere Werte aufweist. Die Positionsgewichtung erfolgt mit der geänderten Symbolverteilung. Es wird ein zweiter Hashwert ermittelt. Die beiden Teil-Hashwerte werden zusammengefaßt, sei es durch Addition oder eine sonstige lineare Kombination und es ergibt sich ein neuer Teil-Hashwert.

Es wird mit der gesamten Datei so weiter verfahren. Weitere Teil-Hashwerte können mit dem sich zuvor ergebenden Teil-Hashwert zusammengefaßt werden, so daß sich inkrementell ein abschließender Hashwert ergibt, der möglichst eindeutig der Klartext-Datei zugeordnet ist (Anspruch 2).

Die Symbolverteilung kann neben einer Steuerung nur durch den kurzen Block zusätzlich einer sich nur mit langer Periode wiederholenden Verschiebung oder Aufschaltung verändern, so daß selbst Dateien die aus nur gleichen Buchstaben bestehen, nicht zu immer gleichen Teil-Hashwerten führen. Eine Verschiebung oder ein Offset nach jeder Symbolverteilung und anknüpfend an einen jeweiligen neuen Block aus der Datei, deren Hashwert zu bilden ist, bietet zusätzliche Sicherheit (Anspruch 7).

Ist die Datei von einer solchen Länge, die kein ganzzahliges Vielfaches der kurzen Blocklänge aufzunehmen vermag, so wird die Datei um entsprechende "Character" ergänzt, bis auch der letzte Block vollständig ist. Diese Ergänzung von Charactern ist unkritisch, es sollte aber sichergestellt sein, daß alle Character der Datei mit der Hashfunktion bearbeitet worden sind, um auszuschließen, daß gerade in dem letzten Block, der nicht vollständig ist und vielleicht unberücksichtigt sein könnte, Änderungen erfolgen, die von dem Hashwert nicht erfaßt sind. Derjenige, der die Hashfunktion erneut anwendet auf die übertragene Datei würde ebenso vorgehen und am Ende eine bekannte Ergänzung hinzunehmen, so zum Beispiel die Zahl "Null" in entsprechender Länge bis zur Auffüllung des letzten Blocks.

Ist die erfindungsgemäß verwendete Hashfunktion öffentlich, so wird sie zur Integritätsprüfung verwendet. Ist sie "privat", also nur dem Sender und Empfänger bekannt, so kann eine Authentizitäts-Kontrolle vorgenommen werden, der Empfänger kann bei nur zweiseitig bekannter Hashfunktion sicher sein, daß die ihm übermittelte Nachricht von demjenigen kommt, der sowohl die Datei abgesendet hat, als auch mit der nur auf beiden Seiten vorhandenen privaten Hashfunktion den Hashwert aus dieser Datei gebildet und mit der Datei übertragen hat (Anspruch 9).

Bei sehr umfangreichen Dateien kann der Hashwert eine sehr große Zahl erreichen, es wird zweckmäßig deshalb in einem höheren Zahlensystem der Hashwert ausgegeben (Anspruch 5, 8).

Eine mehrfache Verwendung bei den Einzelstufen, also bei der Bestimmung der Teil-Hashwerte von jeweils einem kurzen Block, kann integriert werden (Anspruch 6). Jeder Teil-Hashwert setzt sich dann aus zwei, drei oder vier (oder mehreren) anteiligen Hashwerten zusammen, die gemeinsam zusammengefaßt, z. B. addiert, den Teil-Hashwert bilden, der z. B. ein vielfach mit vier verschiedenen Symbolverteilungen in der Matrix ermittelt aus vier anteiligen Hashwerten bestehender Teil-Hashwerte ist. Die Sicherheit dieses Teil-Hashwertes und damit die Sicherheit des aus ihm zusammengesetzten abschließenden Hashwertes nimmt mit zunehmender Zahl der Matrizen zu, wobei hier sowohl ein paralleles Arbeiten von mehreren Matrizen gleichzeitig, die jeweils eine unterschiedliche Symbolverteilung haben, möglich ist, als auch ein sequentielles Arbeiten, bei dem dieselbe Matrix nacheinander mit mehreren unterschiedlichen Symbolverteilungen belegt wird und dann für jede Symbolverteilung die Positionsgewichtung durchgeführt wird, um den jeweiligen anteiligen Hashwert zu errechnen.

Beispiele erläutern und ergänzen die Erfindung.

Fig. 1 veranschaulicht die Ermittlung eines Hashwertes oder einer Hashzahl (Kontrollwort) in der Ausgabeebene 130 unter Berücksichtigung von Klartext-Daten 110 und vorgegebenen externen Schlüsseldaten 100, die von einem Datenträger gespeist werden.

Fig. 2 veranschaulicht ein einfaches Beispiel einer Positionsgewichtung an einer Matrixbelegung mit einer 2×2 Matrix und vier Charactern. Der Teil-Hashwert beträgt 42.

Fig. 3 ist eine detaillierte Darstellung der Fig. 1 mit zwei parallelen Matrizen und einer gesteuerten Symbolverteilung SV.

Fig. 4 veranschaulicht die Blöcke B1, B2, ..., einer Klartext-Datei 110 und ihre Einspeisung in die Symbolverteilung 101.

In der Fig. 1 wird global erläutert, wie der Ablauf der Anwendung einer Hash-Funktion auf Klartext-Daten in einer Datei 110 erfolgt. Ein mathematischer Reihengenerator MRG 101 wird gespeist von einer Karte 100 und den Klartext-Daten der Datei 110. Ein Block von einer bestimmten Anzahl von Zeichen – bevorzugt 40 Byte – aus der Datei 110 bildet den Eingabeblock in Schritt 1. In Schritt 2 wird eine Kontrollzahl K aus der Summe aller ASCII-Codes des Eingabeblocks gebildet. Es wird in Schritt 3 eine Basisreihe B von mindestens 80 Ziffern aus den ASCII-Codes des Eingabeblocks unter Modifizierung der Code-Zahlen durch Subtraktion oder Addition mit einem als internem Bestimmungsfaktor festgelegten konstanten Wert D und Weglassen der ersten Ziffer im Falle von dreistelligen Ergebnissen im Falle der Modifikation gebildet. Die internen Bestimmungsfaktoren iB sind aus Fig. 3 ersichtlich, wo sie in dem MRG als mathematischer Reihengenerator 101 vorgesehen sind, der später erläutert wird.

In Schritt 4 und 5 werden – zuvor, danach oder gleichzeitig – zumindest acht Argumente a mit bis zu 18 Stellen unter Verwendung von Daten einer Stringvariablen S von der Karte 100 geladen. Es werden außerdem mindestens acht Funktionszahlen f aus den Daten der Stringvariablen S von der Karte 100 in den MRG 101 geladen. Dies sind die externen Bestimmungsfaktoren eB, ersichtlich aus Fig. 3.

In Schritt 6 erfolgt eine Berechnung von mindestens acht Ergebnissen mit jeweils mindestens 10 Stellen nach dem Komma unter Anwendung verschiedener mathematischer Funktionen, was die Betriebsweise des MRG 101 beschreibt, der später gesondert erläutert wird.

Der MRG bildet Reihen A, B, C, D von jeweils mindestens 160 Ziffern, wobei so viele Reihen generiert werden, wie Matrizen vorhanden sind, in denen Symbole aus einem Symbolvorrat SM (Symbolmenge) 104 verteilt werden müssen. Diese Verteilung übernimmt die Steuerung SV.

Die Matrizen werden vor Belegung in ihren Dimensionen festgelegt. Hier wird vorgeschlagen, ein internes Zahlensystem Z zur Basis zwischen zwei und höchsten 128 als Maßstab für die Elemente der jeweiligen Matrix zu verwenden. Die Matrizen sind mit mindestens zwei und höchstens acht Dimensionen ausgestattet und haben die Aufgabe die Symbole,

die auf sie verteilt werden, zu speichern.

Es werden in Schritt 10 Folgen der Ziffer Null bis zur höchsten Ziffer nach der vorhergehenden Festlegung des internen Zahlensystems Z gewählt, so daß jede Ziffer des Zahlensystems Z nur einmal vorkommt, aber abhängig von dem MRG frei verteilt sind; doppelt auftretende Ziffern werden ausgeblendet. Das sind dann die Folgen A, B, C und D.

Der Symbolvorrat oder die Symbolmenge 104 gemäß Fig. 3 wird in dem Umfang festgelegt, wie Speicherkapazität in den jeweiligen Matrizen A, B, C und D, hier mit MAT-A und MAT-B repräsentiert und ohne Darstellung von MAT-C, MAT-D, entsprechend den Matrizen A, B, benötigt wird. Die entsprechende Menge ergibt sich aus den Dimensionen, den Zeilen und Spalten, und zwar in der Form, daß jedes Element in der Matrix mit einem Symbol zu belegen ist, so daß die gesamte Matrix kein Symbol doppelt enthält.

Die Verteilung SV erfolgt gesteuert von dem MRG 101, der von den Klartext-Zeichen, respektive dem aus der Datei 110 entnommenen Block B1 (siehe dazu Fig. 4) gesteuert wird, so daß die Klartextzeichen der der Hashfunktion unterworfenen Datei die Verteilung des Symbolvorrates auf die Matrizen MAT-A und MAT-B bestimmen, also steuern, und so steuern, daß jeweils eine Matrix kein Symbol doppelt hat.

Nach dieser unregelmäßigen Verteilung (Permutation) werden Positionsgewichtungen PG durchgeführt, von einem Gewichter 106, der die verteilten Symbole entsprechend ihrem Wert und ihrer Position gewichtet und einen Teil-Hashwert Hash 1 (#1) bildet. Dies kann geschehen durch Multiplikation der ASCII-Codes eines jeden Symbols aus dem vorhergehenden Schritt der Permutation für jede der Matrizen MAT-A und MAT-B mit einer aufsteigenden, numerischen Variablen. Es kann aber ebenso eine fortlaufende Ziffer der zuvor erwähnten Reihen A, B, C und D sein.

Für einen weiteren Block B2 aus der Datei 110 wird eine andere Symbolverteilung vorgenommen, wie sich ersichtlich aus der für den geänderten Klartextblock geltenden Verteilungsvorschrift über den MRG ergibt, der den Symbolvorrat SM auf dieselbe Matrix MAT-A oder eine zweite Matrix MAT-B neu verteilt. Auch dabei findet eine Positionsgewichtung PG statt, die zu einem zweiten Hashwert Hash 2 (#2) führt, der zum ersten Hashwert #1 addiert 102 wird und eine Summe Hash S (#S) bildet. Dieser Summenwert ist der Ausgangswert 130 und wenn alle Blöcke B_i ($i = 1 \dots n$) der Datei 110 über die Symbolverteilung SV in einen jeweiligen Teil-Hashwert abgebildet worden sind, ergibt sich der abschließende Hashwert 130, der der "Fingerabdruck" der Datei 110 ist.

In Fig. 2 ist einfach erläutert, wie die Positionsgewichtung PG arbeitet, ausgehend von der Symbolverteilung 102a in einer als Beispiel herangezogenen Matrix mit vier Elementen. Vier Symbole 8, 4, 2, 5 sind dort verteilt und gewichtet werden sie mit den Zahlen 1, 2, 3 und 4 wie von dem Positionsgewichter 106 angedeutet. Als Hashwert ergibt sich für die erste Teil-Hashzahl der Wert 42.

Die Fig. 4 veranschaulicht, wie eine gesamte Datei 110, die in Blöcken B1, B2, B3 ... (B_i ; $i = 1 \dots n$) mit jeweils 40 byte dargestellt ist, nacheinander eine Symbolverteilung über den MRG 101 und die Symbolverteilung 102 vornimmt und zu weiteren Teil-Hashwerten führt, die hier symbolisch mit Hash 3 (#3) bezeichnet sind. Angedeutet ist ein Multiplexer, der die Leitung 110a jeweils um einen Block B_i weiterschaltet, wenn der vorhergehende Block der Datei 110 der Hashfunktion unterworfen wurde. Am Ende der Datei kann es sein, daß ein Restblock B_n übrigbleibt, der keine 40 byte mehr hat, so daß ein Ergänzungs-Block E angefügt wird, um einen abschließenden Block von wieder 40 byte Länge zu erhalten. Damit ist die gesamte Datei mit allen Klartextzeichen der Hashfunktion unterworfen und jedwede Änderung nach der Durchführung der Funktion ergibt einen anderen Hashwert als Kontrollwort und die Datei ist mit dem berechneten Kontrollwort eindeutig identifiziert, im Sinne eines Fingerabdrucks.

Eine Veränderung der Symbolverteilungssteuerung 102 führt dazu, daß ein Block der Datei 110 mehrere Matrizen MAT-A, MAT-B parallel mit unterschiedlichen Symbolverteilungen oder aber dieselbe Matrix nacheinander mit unterschiedlichen Symbolverteilungen belegt wird und daraus jeweils ein anteiliger Teil-Hashwert gebildet wird, der zunächst in mehreren Durchläufen, wie beispielsweise vier Durchläufe zu einem Teil-Hashwert #1 zusammengesetzt wird. Dadurch steigt zwar die Rechenzeit an, kann aber durch parallele Verarbeitung und mehrere parallele Matrizen, die gleichzeitig belegt werden, und demgemäß auch mehrere parallele Symbolverteiler 102 wieder beschleunigt werden. Die Sicherheit kann durch diese Mehrfachdurchläufe erhöht werden.

Die Teil-Hashwerte werden entweder nach einer jeweiligen Positionsgewichtung oder nach Durchführen aller Positionsgewichtungen auf einmal in dem Addierer 103 addiert und bilden das abschließende Kontrollwort Hash S (#S).

Der MRG 101 ergibt sich aus der folgenden Erläuterung. Ziel sind n-dimensionale Matrizen in einem beliebigen Zahlensystem (Basis 2 bis Basis 128), in denen eine vom gewählten Zahlensystem Z abhängige Menge von ASCII-Zeichen, die eine Teilmenge der ASCII-Zeichen, alle ASCII-Zeichen oder Kombinationen aus zwei oder mehr ASCII-Zeichen als Symbole umfassen, unregelmäßig verteilt enthalten sind. Dabei ist jedes Symbol in jeder Matrix MAT-A, MAT-B, MAT-C oder MAT-D nur einmal vorhanden. Die Verteilung der Elemente in den Matrizen ist abhängig von dem Inhalt der Blöcke der Textdatei 110. Als Kontrollzahl K ermittelt das Programm zunächst die Summe aller ASCII-Werte der Zeichen eines Blocks (hier als Beispiel 40 Zeichen) der Textdatei 110. Weiter wird vom ASCII-Wert jedes einzelnen Zeichens die Konstante $D = 20$ abgezogen (Teil der iB) und bei verbleibenden dreistelligen Zahlen wird die erste Ziffer weggelassen, so daß sich folgende Basisreihe B als Beispiel ergibt:

```
3128333480818401892828362930328728353737
5791979096124998819481959612121237373737
```

Die aus den Textblöcken B_i gewonnene Basisreihe B ist Grundlage für weitere Schritte der Symbolverteilung. Als nächster Schritt werden von der externen Karte 100 eine vom Hersteller mit einem Zufallsgenerator erzeugte und auf der Karte gespeicherte Stringvariable S geladen, zum Beispiel:

```
S = "1468243612435277334932152851567243634917"
```

Die geladene Stringvariable S enthält Faktoren zur Steuerung des mathematischen Reihengenerators MRG. Die ersten

DE 198 11 593 C 1

8 zweistelligen Ziffern 14, 68, 24, 36, 12, 43, 52, 77 zuzüglich jeweils einer im MRG festgelegten Konstanten (hier 1) bestimmen die Positionen, an denen aus der Basisreihe B jeweils drei zusammenhängende Ziffern zuzüglich der Kontrollsumme (hier: 89) als Argumente "a" für den mathematischen Reihengenerator entnommen werden. Mit Hilfe verschiedener mathematischer Funktionen, deren Ergebnisse mindestens 10 Stellen nach dem Komma aufweisen (z. B. SIN, COS, TAN, Log(n), LN, $n^{(1/n)}$) erzeugt der MRG unregelmäßige Reihenfolgen der Ziffern 0 bis 9 mit einer Länge von z. B. jeweils 160 Ziffern. Es werden so viele Reihenfolgen (Reihe A, Reihe B, Reihe C, Reihe D) generiert, wie Matrizen vorhanden sind. Zur Erzeugung der Reihenfolgen werden die Ergebnisse der mathematischen Funktionen aus den Argumenten "a" (Schritt 4) noch mit je einer Funktionszahl f (Schritt 5) multipliziert (alternativ: dividiert, radiziert oder potenziert). Diese Funktionszahlen f entnimmt der MRG z. B. beginnend an den letzten 24 Ziffern der von der Karte 100 geladenen Stringvariablen S vom Ende her gerechnet. Ein Beispiel soll dargestellt werden:

Position der Basisreihe	3 Ziffern an der Position	Argument + 89	Funktionszahl
15	018	107	334
69	948	210	932
25	293	382	152
37	373	462	851
13	840	929	567
44	197	286	243
53	499	588	634
78	737	826	917

Der MRG 101 errechnet in Schritt 6 die einzelnen Ziffernfolgen mit Hilfe der mathematischen Funktionen unter Anwendung der Argumente a und Funktionszahlen f wie folgt:

Ziffer (1) =	334 * LOG10 (107)	=	677,8141817468
Ziffer (2) =	932 * SIN (210)	=	435,9136590954
Ziffer (3) =	152 * 382 ^(1/3)	=	1102,8879090504
Ziffer (4) =	851 * LN (462)	=	5221,3657223105
Ziffer (5) =	567 * COS (929)	=	347,3553802675
Ziffer (6) =	243 * TAN (286)	=	28,08571953232
Ziffer (7) =	634 * 588 ^(1/4)	=	3122,004965034517
Ziffer (8) =	917 * 826 ^(1/2)	=	26354,7778211086

Die Ziffern vor dem Komma werden unterdrückt und alle 10 Ziffern nach dem Komma zu einer Reihenfolge von 80 Ziffern zusammengefaßt. Für eine vollständige Reihe von 160 Ziffern durchläuft der MRG die mathematischen Funktionen ein zweites Mal, jedoch mit anderen Werten. Es werden in mehreren Durchläufen so viele Reihen erzeugt, wie Matrizen MAT-A, MAT-B, MAT-C, MAT-D vorhanden sind. Bei jedem Durchlauf im MRG werden die Argumente a und Funktions-Zahlen f variiert, so daß immer verschiedene und voneinander unabhängige Reihen entstehen. Die Variation wird durch interne Bestimmungsfaktoren iB gesteuert, die vom Hersteller für jedes Programm unterschiedlich festgelegt werden können. Aus den Nachkommastellen obiger Ziffern (1) bis (8) entstehen folgende Reihen.

Reihe A

8141817468913659095488790905043657223105
 3553802675857195323249650345177778211086
 1836559192864814921821893171732243102880
 2503378684519426655113008140125586840054

Reihe B

9205533367818823835481756496791891514967
 6300970849702019082337074467341592950056
 8954855945517351543032005986142809850231
 7267581504683709464990740885388475569671

Reihe C

2797769253577970185835493859941433652582
 8700812331636689811564135562932009966596
 1759393111869064060239489931956496715430
 8567793727595004749378336296189416895708

Reihe D

6913387348849361427491144516131055677507
 2282257574424203035333950581608209701734
 7364245858963322560584198758839319881682
 4197677174450678683095727991529216679944

Die Verteilung der Symbole auf die Zeilen und Spalten einer der Matrizen MAT-A, MAT-B schafft eine gründliche Durchmischung der ursprünglich geordneten Menge aller ASCII-Zeichen (Permutation) die von der Textdatei gesteuert ist. Um das zu erreichen, werden aus den Reihen A, B (C, D) Folgen der Ziffern 0 bis zur höchsten Ziffer im gewählten Zahlensystem (hier zur Basis 15 die Ziffern 0 bis 14) abgeleitet, in welchen Folgen jede Ziffer nur einmal vorkommt. Diese Folgenerzeugung, deren Ergebnis unten dargestellt ist, baut auf den vorherigen Reihen auf.

Es soll grob skizziert werden, daß interne Bestimmungsfaktoren iB eine Steuergröße für die Auswahl bestimmter Bereiche darstellen, aus denen zweistellige Wertepaare nacheinander entnommen werden und so verarbeitet werden, daß keine Ziffer doppelt vorkommt. Eine Möglichkeit liegt darin, die entnommene Wertepaare in einer Modulo-Berechnung auf solche Wertepaare zu reduzieren, die sich im gewählten Zahlensystem halten, hier also Modulo 15 zur Abbildung jedes Wertepaares in eine Ziffer 0 bis 14; entsteht bei sukzessiver Auswertung des ausgewählten Bereiches eine Zahl erneut, die zuvor bereits ermittelt war, so wird sie übersprungen und zum nächsten Wertepaar übergegangen. Ist der Bereich, der beispielsweise 20 Ziffern umfassen kann, abgearbeitet, so bestimmt ein interner Bestimmungsfaktor einen neuen Anfangspunkt in der Reihe, von dem ausgehend ein zweiter Bereich von beispielsweise auch 20 Ziffern herausgenommen wird, dessen Wertepaare sukzessive mit der Modulo-Berechnung und dem Vergleich, ob die modulo-berechnete Zahl schon auftrat, unterworfen wird. Ein Beispiel für vier ermittelte Folgen A bis D aus den vier Reihen A bis D ist unten angegeben. In einer jeweiligen Folge tritt kein Wert doppelt auf.

Folge A:	4	13	5	6	7	8	9	3	10	2	11	12	14	1	0
Folge B:	4	10	2	5	0	13	3	6	7	8	1	12	9	11	14
Folge C:	11	3	9	6	10	0	8	4	14	5	7	12	13	1	2
Folge D:	10	14	3	2	11	0	13	12	8	1	4	9	5	6	7

Da jede Folge die Durchmischung der Zeichen in der zugeordneten Matrix steuert, werden mindestens so viele unterschiedliche Folgen erzeugt, wie Matrizen vorhanden sind. Sowohl die Reihen als auch die Folgen ändern sich, sobald auch nur ein Wert in dem Textblock aus der Datei 110 geändert wird. Damit sind die Verteilungen der Symbole in den Matrizen abhängig von dem Inhalt der Blöcke der Textdatei. Der Hashwert ist über die PC-Matrix gesteuert.

1. Verfahren zum Berechnen eines Kontrollwortes (Hashwert) einer Text- oder Informationsdatei, das mit der Datei zusammen übertragen wird, um dem Empfänger die Integrität der mitübertragenen Datei oder die richtige Herkunft (Authentizität) der Datei zu signalisieren, oder ohne die Datei als Kontrollwort alleine übertragen wird, um dem Empfänger einen Vergleichswert zur Verfügung zu stellen, so daß eine beim Empfänger vorhandene Datei mit einer nicht mitübertragenen Datei über das Kontrollwort verglichen werden kann, bei welchem Verfahren
 - (a) mit einem ersten, in seiner Länge gegenüber der Gesamtlänge der Datei (110) kurzen Block von Zeichen (B1) der Datei eine erste Symbolverteilung (SV, 102) in einer eine vorgegebene Dimension aufweisenden Matrix (MAT-A, MAT-B) vorgenommen wird, so daß jedes Element der Matrix ein anderes Symbol aus einem Symbolvorrat (SM) enthält;
 - (a1) die erste Symbolverteilung in der Matrix positionsgewichtet (106, PG) zu einem ersten Teil-Hashwert (#1) zusammengefaßt wird;
 - (b) mit einem weiteren, der Länge des ersten Blocks entsprechenden zweiten Block (B2) von Zeichen derselben Datei (110) eine geänderte Symbolverteilung (SV, 102) in derselben Matrix (MAT-A) oder einer zweiten Matrix (MAT-B) vorgenommen wird;
 - (b1) die geänderte Symbolverteilung erneut positionsgewichtet (106) wird, um einen weiteren Teil-Hashwert (#2) zu errechnen;
 um dann beide Teil-Hashwerte (#1, #2) gleich zusammenzufassen oder später nach Berechnen mehrerer oder aller Teil-Hashwerte zu einem abschließenden Hashwert (#S) zusammenzufassen.
2. Verfahren nach Anspruch 1, bei dem die gesamte Datei (110) nacheinander in Teil-Hashwerte umgerechnet wird, deren Gesamtsumme als abschließender Hashwert (#S) mit der Übertragung der nicht verschlüsselten Datei als Integritätsnachweis an den Empfänger übertragbar ist, oder alleine als Hashwert ohne Übertragung der Datei an einen Empfänger zu Vergleichszwecken übermittelbar ist oder von einem Empfänger als Vergleichswert von einer empfangenen Datei errechnet wird, um mit dem mitübertragenen abschließenden Hashwert (#S) derjenigen Datei (110) vor der Übertragung am Empfängerort verglichen zu werden.
3. Verfahren nach einem der vorigen Ansprüche, bei dem die Länge eines Blocks unter 100 byte und über 2 byte, insbesondere etwa zwischen 30 und 50 byte liegt.
4. Verfahren nach einem der vorigen Ansprüche, bei dem die Symbole der Symbolverteilung (SV) ASCII-Worte oder -Character sind.
5. Verfahren nach einem der vorigen Ansprüche, bei dem die Teil-Hashwerte im Zahlensystem zumindest zur Basis 8 definiert und ausgegeben werden.
6. Verfahren nach einem der vorigen Ansprüche, bei dem die Bildung eines Teil-Hashwertes aus einem der kurzen Blöcke mehrfach wiederholt wird, insbesondere zwei- bis viermal, jedes Mal mit einer geänderten Symbolverteilung in der Matrix oder gleichzeitig mit mehreren verschiedenen Symbolverteilungen in mehreren parallelen Matrizen und jede Symbolverteilung positionsgewichtet (106) wird, um einen jeweils anteiligen Hashwert an dem Teil-Hashwert zu ergeben, wobei eine Zusammenfassung der anteiligen Hashwerte den Teil-Hashwert (#1, #2) ergibt.
7. Verfahren nach einem der vorigen Ansprüche, bei dem in die Symbolverteilung (SV) ein Modulo-Verhalten mit langer Periode integriert ist.
8. Verfahren nach einem der vorigen Ansprüche, bei dem der endgültige Hashwert (#S) in einem höheren Zahlensystem ausgegeben wird, als das Zahlensystem der Blöcke ($B_i, i = 1 \dots n$) der Datei (110).
9. Verfahren nach einem der vorigen Ansprüche, bei dem
 - (a) die Verteilungsvorschrift (101, 102) der Symbolverteilung (SV) öffentlich ist, um mehreren Empfängern die Möglichkeit zu geben, die Integritätskontrolle durchzuführen;
 - (b) die Verteilungsvorschrift (101, 102) für die Symbolverteilung (SV) der Symbole in der Matrix nur wenigen, insbesondere nur zwei Personen bekannt ist, um dem Empfänger eine Authentizitätskontrolle der übermittelten Datei oder anhand des übermittelten Hashwertes zu ermöglichen.
10. Verfahren – insbesondere nach Anspruch 1 –, bei dem der Empfänger eines Kontrollwortes (Hashwert, #S) einer Text- oder Informationsdatei, das mit der Datei zusammen übertragen wird, oder ohne die Datei als Kontrollwort alleine übertragen wird, eigenständig ein Kontrollwort errechnet, bei welchem Verfahren
 - (a) mit einem ersten, in seiner Länge gegenüber der Gesamtlänge der übertragenen oder vorhandenen Datei kurzen Block von Zeichen der Datei eine erste Symbolverteilung (102, 101) in einer eine vorgegebene Dimension aufweisenden Matrix (MAT-A, MAT-B) vorgenommen wird, so daß jedes Element der Matrix ein anderes Symbol enthält;
 - (b) die erste Symbolverteilung in der Matrix positionsgewichtet (PG, 106) zu einem ersten Teil-Hashwert (#1) zusammengefaßt wird;
 - (c) mit einem weiteren, der Länge des ersten Blocks entsprechenden zweiten Block von Zeichen derselben Datei (110) eine geänderte Symbolverteilung (102, 101) in derselben Matrix oder einer zweiten Matrix vorgenommen wird;
 - (d) die geänderte Symbolverteilung erneut positionsgewichtet (PG) wird, um einen weiteren Teil-Hashwert (#2) zu errechnen;
 um dann beide Teil-Hashwerte gleich zusammenzufassen oder später nach Berechnen mehrerer oder aller Teil-Hashwerte zu einem abschließenden Hashwert (#S) zusammenzufassen und diesen Wert mit dem übermittelten Hashwert zu vergleichen.
11. Verfahren nach Anspruch 10, bei dem die gesamte Datei über Teil-Hashwerte in einen abschließenden

DE 198 11 593 C 1

Hashwert am Empfängerort abgebildet wird.

Hierzu 3 Seite(n) Zeichnungen

5

10

15

20

25

30

35

40

45

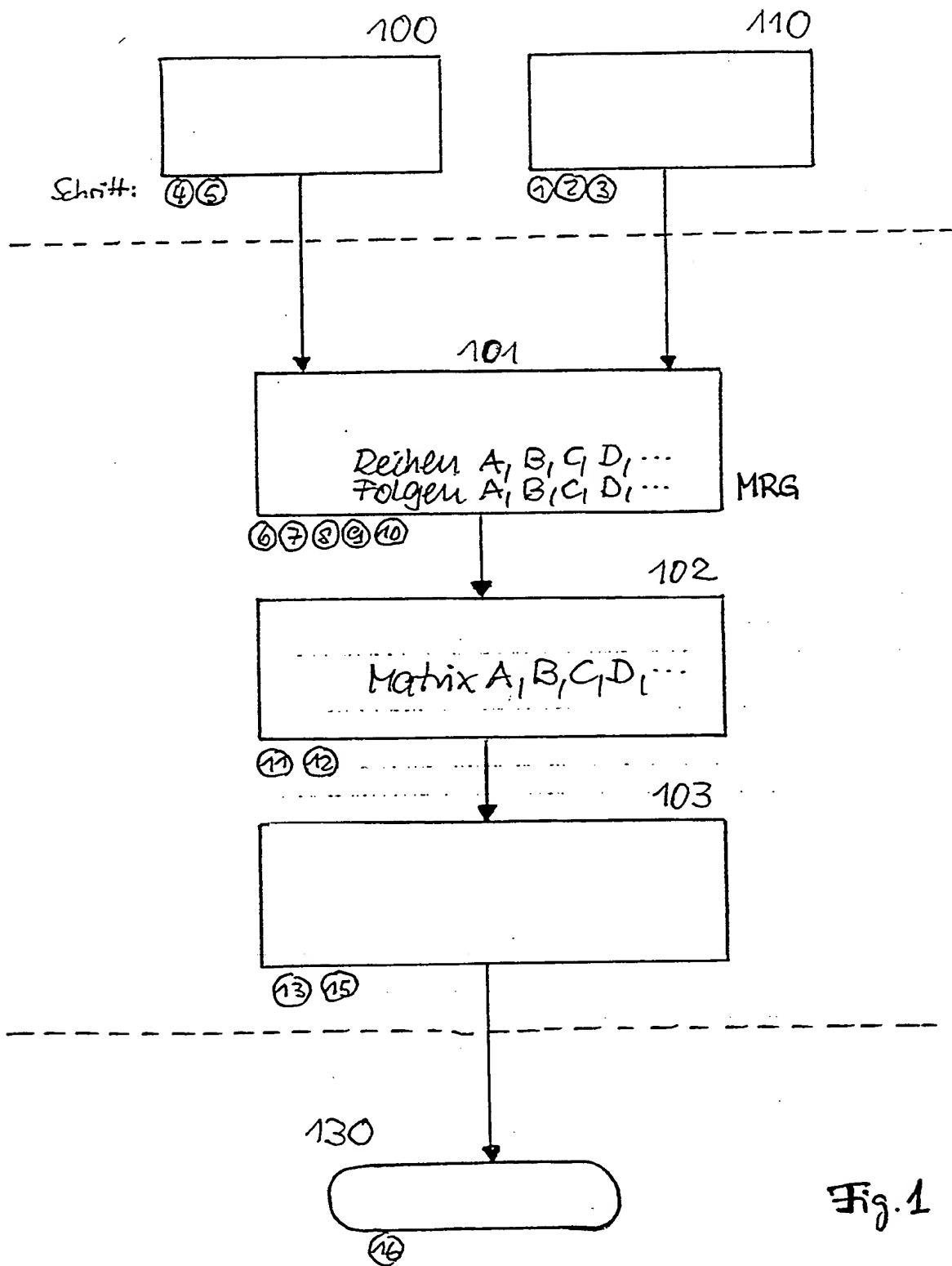
50

55

60

65

- Leerseite -



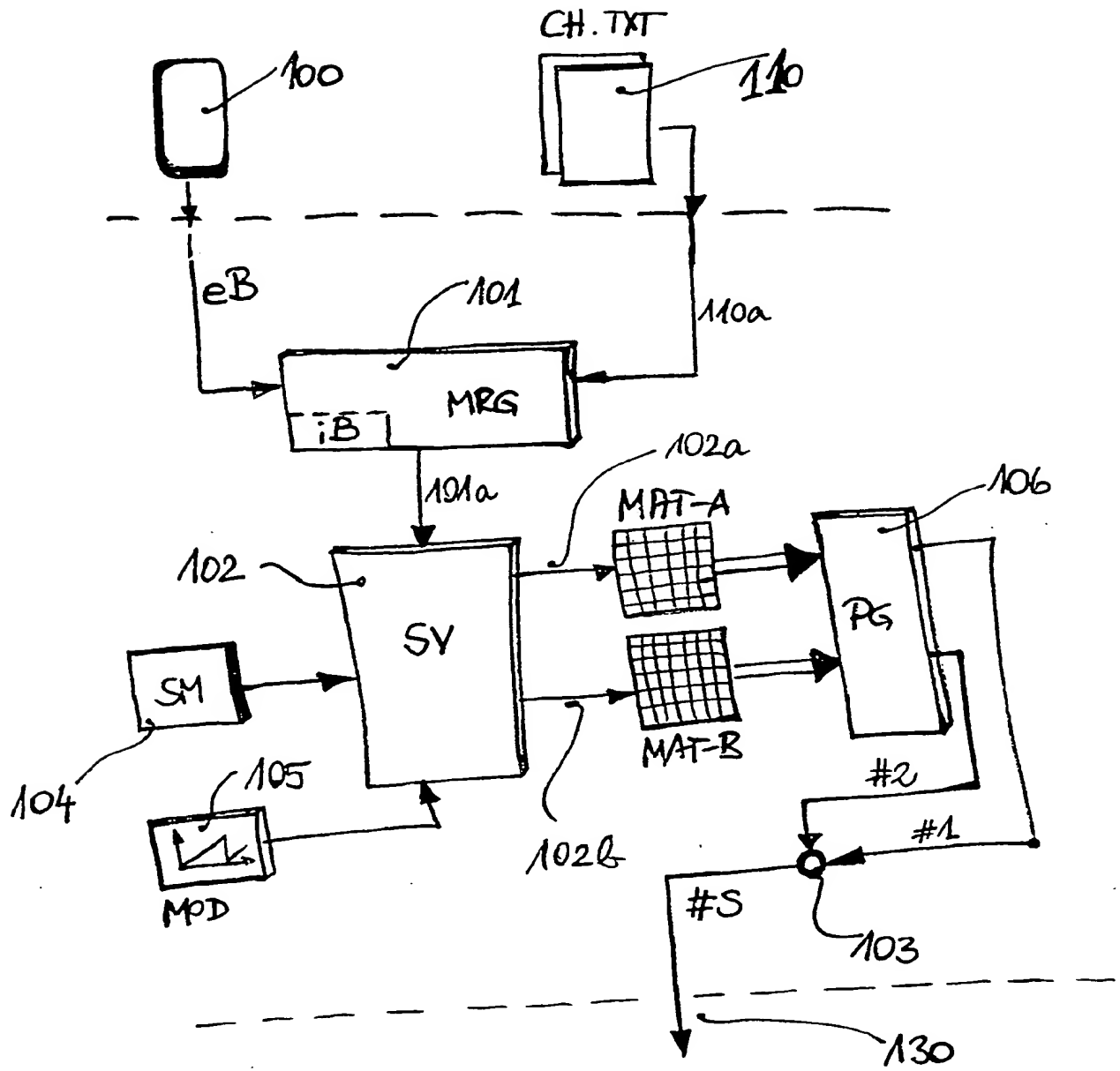


Fig. 3

